

# Sistemet e Bazës së të Dhënave

## SQL DML: JOINS

---

Ramiz HOXHA

ramiz.hoxha@ubt-uni.net

2020/2021

FAKULTETI: SHKENCAVE KOMPJUTERIKE DHE INXHINIERIS

# Objektivat e ligjërates

---

## Në këtë ligjerata, do të mësojnë:

- Komandat DML (Gjuhës për Manipulimi e të Dhënave)
- Kujtim: nderlidhjen (relacionet) e tabelave sipas produktit kartizian
- nderlidhjen (relacionet) e tabelave sipas JOINS

Studenti				
Student_ID	Emri	Mbiemri	Gjinia	Qyteti
S100	Liberta	Buja	F	PZ
S101	Berat	Gashi	M	FR
S102	Edita	Syla	F	PZ
S103	Visar	Dema	M	GJK
S104	Kushtrim	Hoxha	M	PR
S105	Lenonora	Peci	F	PR
S106	Kushtrim	Hoxha	M	PR
S107	Lenonora	Peci	F	PR
S108	Besiana	Shkambi	F	PR
S109	Veton	Sopa	M	PZ
S110	Alketa	Jakupi	F	FR

Lenda		
Kodi_L	Emertimi	ECTS
CAO	Arkitektura Kompjuterike dhe Organizimi	5
CN	Rrjetat Kompjuterike	5
CS2	Shkenca Kompjuterike 2 (POO)	7
DBS	Sistemi i Bazave te Dhenave	6
DSA	Alogoritme dhe Struktura e Dhenave	4
SD	Struktura Diskrete 1	5

Vlersimi			
Studenti	Lenda	Emertim	Nota
S107	DBS	Dhjetë	10
S100	CAO	Shtatë	7
S102	CN	Tetë	8
S103	DBS	Nëntë	9
S106	DBS	Tetë	8
S108	CS2	Tetë	8
S100	CS2	Dhjetë	10
S101	CN	Gjashtë	6
S100	CN	Nëntë	9
S102	CAO	Nëntë	9

Studenti			
Column Name	Condensed Type	Nullable	
Student_ID	char(5)	No	
Emri	varchar(20)	Yes	
Mbiemri	varchar(20)	Yes	
Gjinia	char(1)	Yes	
Qyteti	varchar(10)	Yes	

Lenda			
Column Name	Condensed Type	Nullable	
Kodi_L	char(5)	No	
Emertimi	varchar(40)	Yes	
ECTS	numeric(18, 0)	Yes	

Vlersimi			
Column Name	Condensed Type	Nullable	
Nota	int	Yes	
Emertimi	varchar(10)	Yes	
Studenti	char(5)	Yes	
Lenda	char(5)	Yes	

```

create database Ligj08
use Ligj08
create table Studenti(
    Student_ID char(5),
    Emri varchar (20),
    Mbiemri varchar (20),
    Gjinia char(1),
    Qyteti varchar (10),
    Primary Key (Student_ID),
    check (Gjinia IN ('M', 'F', 'N'))
);

```

```

insert into Studenti values ('S100', 'Liberta', 'Buja', 'F', 'PZ');
insert into Studenti values ('S101', 'Berat', 'Gashi', 'M', 'FR');
insert into Studenti values ('S102', 'Edita', 'Syla', 'F', 'PZ');
insert into Studenti values ('S103', 'Visar', 'Dema', 'M', 'GJK');
insert into Studenti values ('S104', 'Kushtrim', 'Hoxha', 'M', 'PR');
insert into Studenti values ('S105', 'Lenonora', 'Peci', 'F', 'PR');
insert into Studenti values ('S106', 'Kushtrim', 'Hoxha', 'M', 'PR');
insert into Studenti values ('S107', 'Lenonora', 'Peci', 'F', 'PR');
insert into Studenti values ('S108', 'Besiana', 'Shkambi', 'F', 'PR');
insert into Studenti values ('S109', 'Veton', 'Sopa', 'M', 'PZ');
insert into Studenti values ('S110', 'Alketa', 'Jakupi', 'F', 'FR');

```

```

create table Lenda(
    Kodi_L char(5) Primary Key ,
    Emertimi varchar (40),
    ECTS numeric
);

```

```

insert into Lenda values ('DBS', 'Sistemi i Bazave te te Dhenave', '6')
insert into Lenda values ('CS2', 'Shkenca Kompjuterike 2 (POO)', '7')
insert into Lenda values ('DSA', 'Alogoritme dhe Struktura e te Dhenave', '4')
insert into Lenda values ('CA0', 'Arkitektura Kompjuterike dhe Organizimi', '5')
insert into Lenda values ('SD', 'Strukturata Diskrete 1', '5')
insert into Lenda values ('CN', 'Rrjetat Kompjuterike', '5')

```

```

CREATE TABLE Vlersimi (
    Nota int,
    Emertimi varchar(10),
    Studenti char(5),
    Lenda char(5),
    FOREIGN KEY (Studenti) REFERENCES
        Studenti(Student_ID)
        ON UPDATE CASCADE ON DELETE SET NULL,
    FOREIGN KEY (Lenda) REFERENCES
        Lenda(Kodi_L)
        ON UPDATE CASCADE ON DELETE SET NULL
);

```

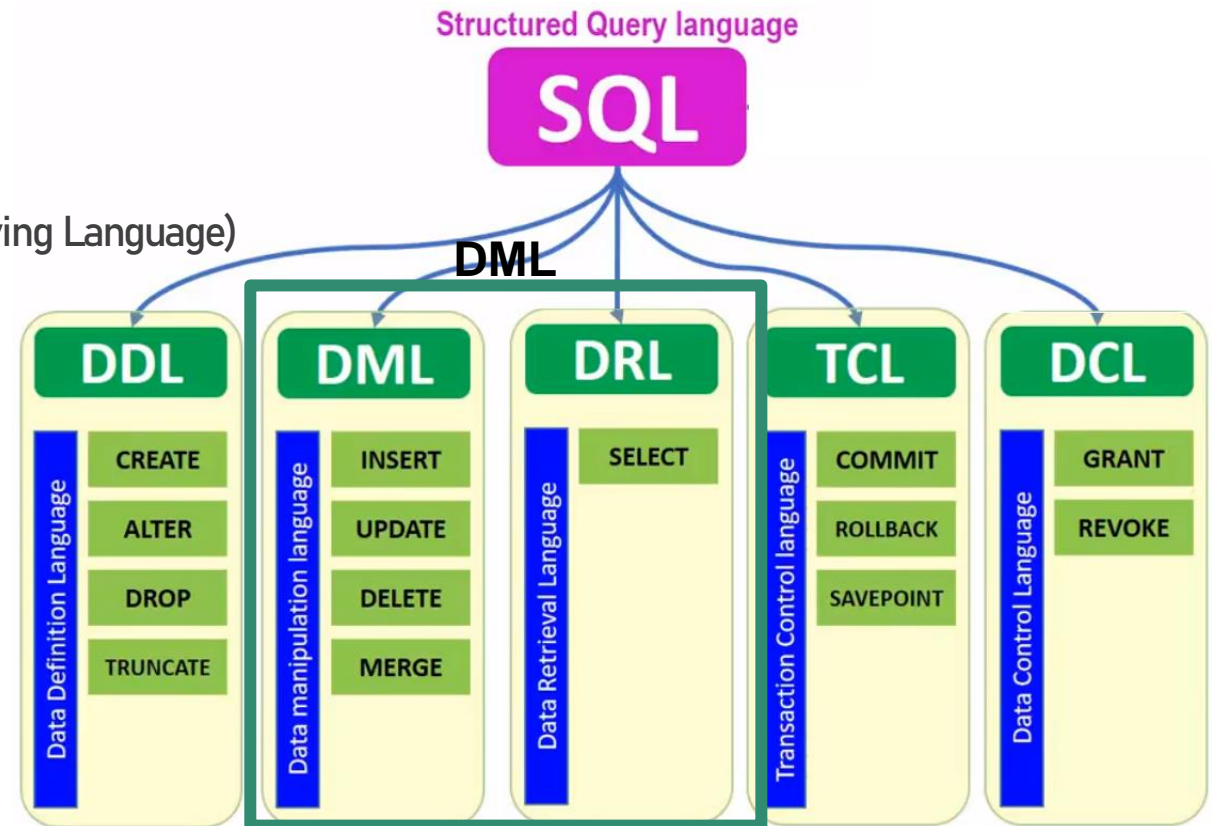
```

Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values
('S102', 'CA0', 'Nëntë', 9)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values ('S100',
'CA0', 'Shtatë', 7)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values ('S102',
'CN', 'Tetë', 8)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values ('S103',
'DBS', 'Nëntë', 9)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values
('S106', 'DBS', 'Tetë', 8)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values
('S108', 'CS2', 'Tetë', 8)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values ('S100',
'CS2', 'Dhjetë', 10)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values
('S101', 'CN', 'Gjashtë', 6)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values
('S100', 'CN', 'Nëntë', 9)
Insert into Vlersimi (Studenti,Lenda,Emertimi,Nota) values
('S107', 'DBS', 'Dhjetë', 10);

```

# Kujtojm: Llojet e deklaratave në SQL

- ❑ **DDL: Gjuha e Definimit të të Dhënave**  
(Data Definition Language)
- ❑ **DML: Gjuha e Manipulimit të të Dhënave**  
(Data Manipulation Language)
  - **DRL: Gjuha për Nxjerrjen e të Dhënave** (Data Retrieving Language)
- ❑ **DCL: Gjuha e Kontrollit të të Dhënave**  
(Data Control Language)
- ❑ **TCL: Gjuha e Kontrollit të Transaksionit**  
(Transaction Control Language)



# Kujtojm: Llojet e deklaratave në SQL

## □ Gjuha e manipulimit të të dhënave (Data manipulation language-DML)

### ■ Përfshin komandat si:

- INSERT, UPDATE, DELETE dhe SELECT

## □ Sintaksa themelore e komandës INSERT duket kështu:

- **INSERT INTO** emri\_tabelës **VALUES** (vlera 1, vlera 2,..., vlera n).

```
insert into Studenti values  
( 'S110', 'Alketa', 'Jakupi', 'F', 'FR' )
```

```
INSERT INTO TABLE (kolona1, kolona2,...)  
SELECT  
    kolona1, kolona2,...  
FROM  
    TABELA_TJETER  
WHERE  
    kushtet
```

```
INSERT INTO shippers_tmp (shipperid,name,phone)  
SELECT  
    shipperid, companyName, phone  
FROM  
    shippers
```

# Updating/Përditësimi i rreshtave të Tabelave

□ Përdorni komandës **UPDATE** për të modifikuar të dhënat në një tabelë.

- **Sintaksa** për këtë komandë është si më poshtë:

```
UPDATE emriTabeles
```

```
SET kolona = shprehja [, kolona = shprehja] ose
```

```
[WHERE kushtet];
```

```
UPDATE tabela1
```

```
SET column1 = (SELECT shprehja1
```

```
FROM tabela2
```

```
WHERE kushtet)
```

```
[WHERE kushtet];
```

- **p.sh:** të modifikoni **ECTS** për lënden **CS2** për këtë komandë është si më poshtë:

```
UPDATE LENDA
```

```
SET ECTS = 6
```

```
WHERE Kodi_L = 'CS2';
```

- **p.sh:** të përditësoni e studentit me ID **'S108'** notën nga 6 në 8 për lënden **CS2**:

```
UPDATE VLERSIMI
```

```
SET Nota = 8, Emertimi='Tetë'
```

```
WHERE Lenda = 'CN' AND Studenti='S101';
```

# Deleting/Fshirja i rreshtave të Tabelave

□ Është e lehtë të fshish një rresht në tabele duke përdorur deklaratën DELETE.

- Sintaksa është si më poshtë:

```
DELETE FROM emriTabeles  
[WHERE kushtet];
```

- p.sh duam të i fshijëm të dhënat që i ndryshuam paraprakisht, të student me ID 'S101' dhe Lenda me kodin e lendes 'CN'

```
DELETE FROM VLERSIMI  
WHERE Lenda = 'CN' AND Studenti = 'S101';
```



# JOINING tabelat e Databazës

## Krijimi e lidhjeve përmes **Çelësit të huaj**

Studenti				
Student_ID	Emri	Mbiemri	Gjinia	Qyteti
S100	Liberta	Buja	F	PZ
S101	Berat	Gashi	M	FR
S102	Edita	Syla	F	PZ
S103	Visar	Dema	M	GJK
S104	Kushtrim	Hoxha	M	PR
S105	Lenonora	Peci	F	PR
S106	Kushtrim	Hoxha	M	PR
S107	Lenonora	Peci	F	PR
S108	Besiana	Shkembi	F	PR
S109	Veton	Sopa	M	PZ
S110	Alketa	Jakupi	F	FR

Çelsi primar

Lenda		
Kodi_L	Emertimi	ECTS
CAO	Arkitektura Kompjuterike dhe Organizimi	5
CN	Rrjetat Kompjuterike	5
CS2	Shkenca Kompjuterike 2 (POO)	7
DBS	Sistemi i Bazave te te Dhenave	6
DSA	Alogoritme dhe Struktura e te Dhenave	4
SD	Strukturura Diskrete 1	5

Produkti kartizian

```

/*Listo studentat e qyteti PR, duke shfaqur lendet te cilat
e kan kaluar me sukses. Duke e paraqitur noten e secilit.*/
SELECT S.Emri, S.Mbiemri, L.Emertimi AS Lenda, V.Nota,
S.Qyteti
FROM Studenti S, Vlersimi V, Lenda L
WHERE s.Student_ID=v.Studenti AND l.Kodi_L=v.Lenda
and S.Qyteti = 'PR'
    
```

Çelsi jashtëm

Çelsi jashtëm

Vlersimi			
Studenti	Lenda	Emertim	Nota
S107	DBS	Dhjetë	10
S100	CAO	Shtatë	7
S102	CN	Tetë	8
S103	DBS	Nëntë	9
S106	DBS	Tetë	8
S108	CS2	Tetë	8
S100	CS2	Dhjetë	10
S101	CN	Gjashtë	6
S100	CN	Nëntë	9
S102	CAO	Nëntë	9

Lidhja ndermjet  
Studentit, vlersimit,  
dhe Lendes

Results Messages					
	Emri	Mbiemri	Lenda	Nota	Qyteti
1	Lenonora	Peci	Sistemi i Bazave te te Dhenave	10	PR
2	Kushtrim	Hoxha	Sistemi i Bazave te te Dhenave	8	PR
3	Besiana	Shkembi	Shkenca Kompjuterike 2 (POO)	8	PR

# Ndërlidhja e Tabelave me JOINS

---

PJESA DYTË

# JOINING tabelat/relacionet e Bazës së të Dhënave

---

- Lidhja mes relacioneve/**JOIN** është e rëndësishme që bënë dallimin në baza të të dhënave relacionale
  - Bashkimi (**JOIN**) kryhet kur dhëna janë marrur nga më shumë se një tabelë njëkohësisht.
  - Për t'i bashkuar tabelat, ju thjesht listoni tabelat me klauzolën **FROM** në deklaratën **SELECT**.
    - SMBDH do të krijojë **produkt kartezipan** i çdo tabele në klauzolën **FROM**.
  - Përdorni *klauzolë* **WHERE** për të *referencuar* atributet që përdoruren për lidhjen e tabelave
    - klauzola **WHERE** nganjëherë quhet si një kusht për bashkim të tabelave.

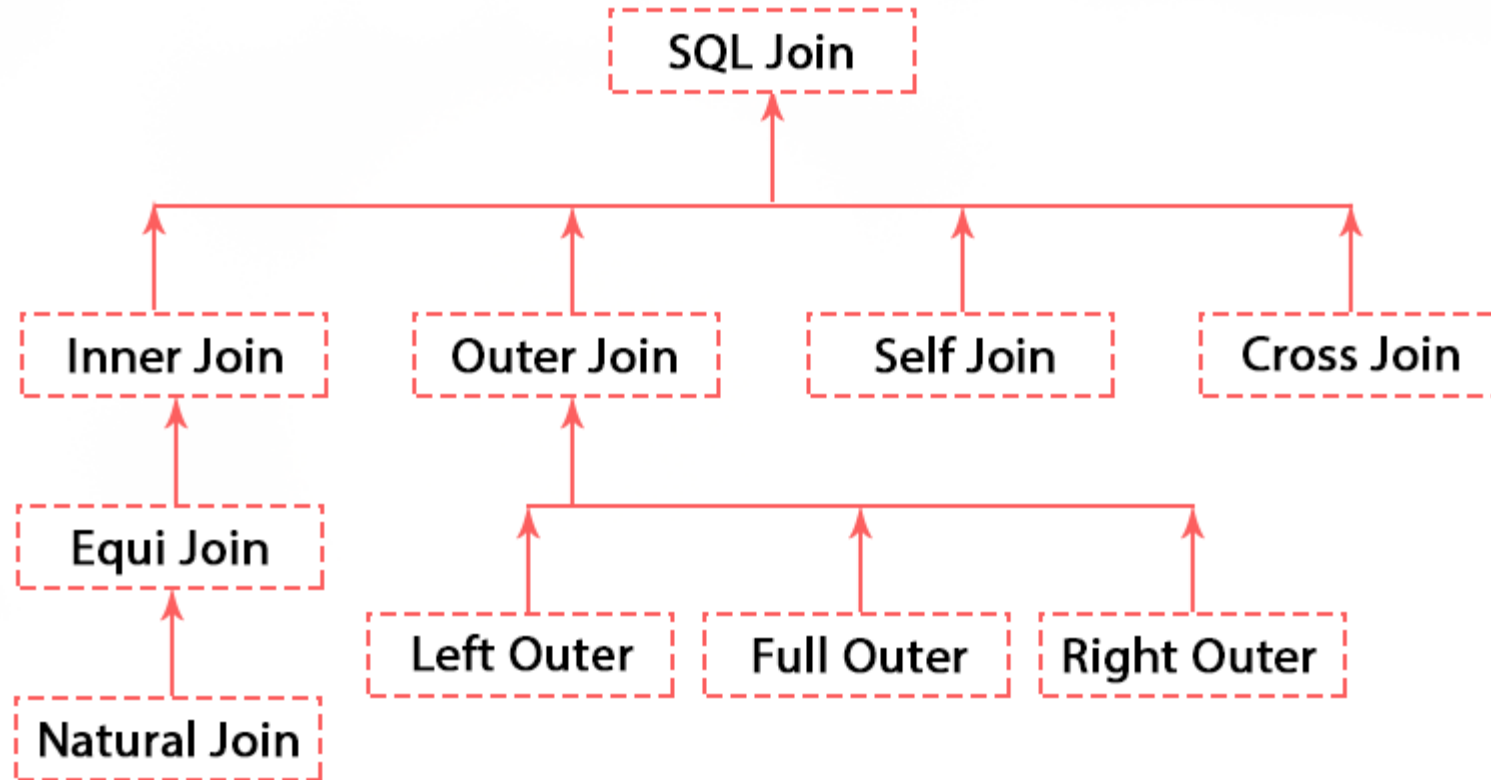
# JOINS

- ❑ Modeli relacional ju mundson të merrni të dhëna nga tabela të veçanta në relacione të reja.
- ❑ Relationet bëhen të qarta kur manipulohen me të dhënat:
  - kur ato i kërkonin në bazën e të dhënave, jo kur e krijoni.
  - Mund të bashkoheni (JOIN) në çdo kolonë në tabela, për sa kohë që llojet e të dhënave përputhen dhe operacioni ka kuptim.
- ❑ JOINS i mirë
  - Kolonat e JOIN duhet të kenë lloje të përputhshëm të të dhënave.
  - Join column is usually key column Kolona e JOIN është zakonisht kolona Çelës:
- ❑ Ose Çelësi kryesor (PK) ose Çelësi i huaj (FK)
  - Nulls kurr nuk do të bashkohen (join).

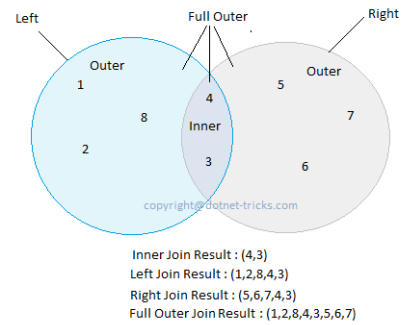
# Operatoret SQL të Bashkimit (JOINS)

- ❑ **Operacionet SQL të bashkimit (JOINS)** përdoret për të kombinuar rreshtat nga **dy** ose më shumë tabelave, bazuar në kolonat lidhse mes atyre tabelave. Më që rast kthejn rreshta sipas një nga kushtet e mëposhtme:
  - Kanë vlera të **përbashkëta** në **kolona** të përbashkëta (bashkohet natyrshëm/natural join).
  - Arrin kushtet të caktuar të bashkimit (barazis ose pabarazis).
  - Kanë vlera të **përbashkëta** në **kolona** të përbashkëta ose nuk kanë vlera të përputhshme (bashkimi i jashtëm).
- ❑ Kemi **4 lloje themelore** të operacionit **JOIN**:
  - **(INNER) JOIN**: Zgjidhni ose selektojn të dhënat që kanë përputhje të vlerave në të dy tabelat.
  - **LEFT (OUTER) JOIN**: Zgjidhni të dhënat nga tabela e parë (të majtës më së shumti) që përputhen me të dhënat e tabelës së djathtë.
  - **RIGHT (OUTER) JOIN**: Zgjidhni të dhënat nga tabela e dytë (djathtes më së shumti) që përputhen me të dhënat e tabelës së majtë.
  - **FULL (OUTER) JOIN**: Zgjedh të gjitha të dhënat që përputhen me cilendo nga të dhënat e tabelave e së majtës dhe e së djathtës.

[https://way2tutorial.com/sql/sql\\_join\\_types\\_visual\\_venn\\_diagram.php](https://way2tutorial.com/sql/sql_join_types_visual_venn_diagram.php)



( SQL JOIN TYPES )



# Lloje të ndryshem të JOIN-nëve

## Llojet e 'JOIN' -neve

### INNER JOIN (d.m.th: JOIN)

- Shembulli i mëparshëm është një **bashkim** i brendshëm (INNER JOIN)

### LEFT OUTER JOIN (d.m.th: LEFT JOIN)

### RIGHT OUTER JOIN (d.m.th: RIGHT JOIN)

### FULL OUTER JOIN (d.m.th: FULL JOIN)

- tab1 **tipi\_join** tab2 ON [ kushti\_join\_ ]
- kushtet e JOIN-it normalisht krahason *dy kolona*, të cilat **duhet** të jenë të të **njëtit lloj të të dhënave**.  
✓ (numri = numri, string = string)
- kryesisht '**Çelësi Primar**' = '**Çelësi i Hua**'

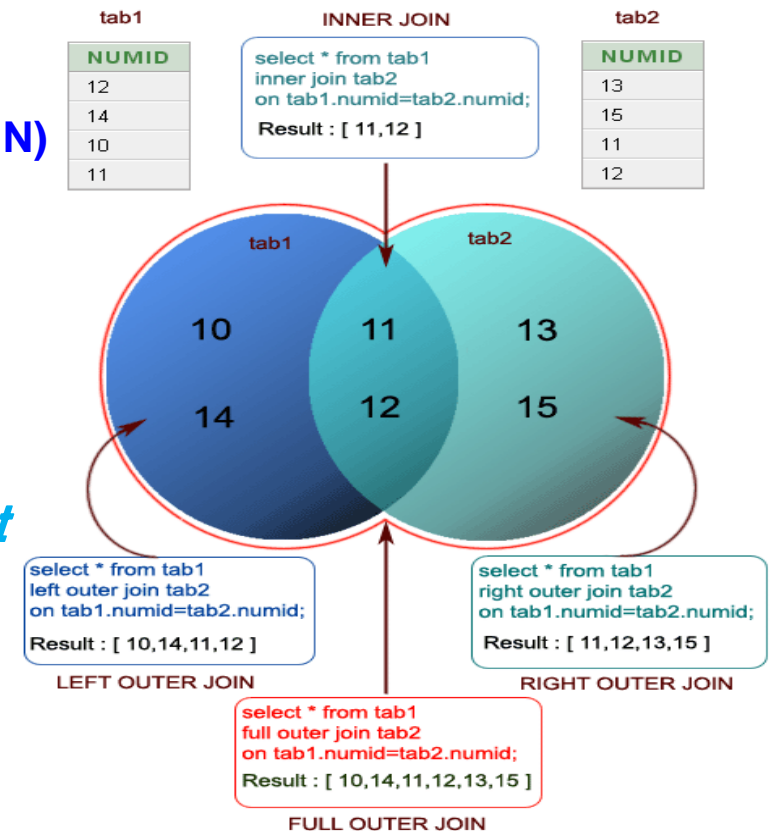


TABLE 8.1

## SQL JOIN EXPRESSION STYLES

JOIN CLASSIFICATION	JOIN TYPE	SQL SYNTAX EXAMPLE	DESCRIPTION
CROSS	CROSS JOIN	SELECT * FROM T1, T2	Returns the Cartesian product of T1 and T2 (old style)
		SELECT * FROM T1 CROSS JOIN T2	Returns the Cartesian product of T1 and T2
INNER	Old-style JOIN	SELECT * FROM T1, T2 WHERE T1.C1=T2.C1	Returns only the rows that meet the join condition in the WHERE clause (old style); only rows with matching values are selected
	NATURAL JOIN	SELECT * FROM T1 NATURAL JOIN T2	Returns only the rows with matching values in the matching columns; the matching columns must have the same names and similar data types
	JOIN USING	SELECT * FROM T1 JOIN T2 USING (C1)	Returns only the rows with matching values in the columns indicated in the USING clause
OUTER	JOIN ON	SELECT * FROM T1 JOIN T2 ON T1.C1=T2.C1	Returns only the rows that meet the join condition indicated in the ON clause
	LEFT JOIN	SELECT * FROM T1 LEFT OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from the left table (T1) with unmatched values
	RIGHT JOIN	SELECT * FROM T1 RIGHT OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from the right table (T2) with unmatched values
	FULL JOIN	SELECT * FROM T1 FULL OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from both tables (T1 and T2) with unmatched values



# INNER JOIN (JOIN i thjeshtë)

□ SQL Server **INNER JOIN** kthehen të gjitha rreshtat nga tabelat të shumta ku kusht i **JOIN** është plotësuar.

- Sintaksa për **INNER JOIN** në SQL Server (krijoj-SQL) është:

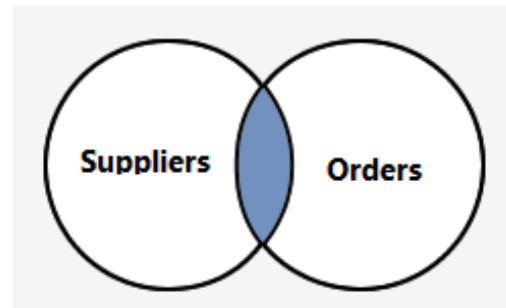
```
SELECT t1.kolona, t2.kolona
FROM tabela1 t1 INNER JOIN
      tabela t2
ON t1.PRIMARYKEY = t2.FOREIGNKEY
```

S_ID	Supplier Name
1000	IBM
1010	Hewlett Packard
1020	Microsoft
1030	NVIDIA

O_ID	Suppliers_ID	Order Date
5000	1000	2003-05-12
5010	1010	2003-05-13
5030	1030	2003-05-14
5040	1000	2003-06-14

## INNER JOIN

```
Select S.S_ID, S.[Supplier
Name], O.O_ID
From Suppliers S
Inner Join Orders O
On S.S_ID =
O.Suppliers_ID;
```



Rezultati: **INNER JOIN**

S_ID	Supplier Name	O_ID
1000	IBM	5000
1010	Hewlett Packard	5010
1030	NVIDIA	5030
1000	IBM	5040

## INNER JOIN (JOIN i thjeshtë)

/\*Në raportin tuaj të shfaqeni të gjithë studentet që kanë kaluar lëndët përkatse. Në këtë raport të shaqet emri, mbiemri, lëndën dhe notën për secilin student.\*/

```
Select s.Emri, s.Mbiemri, l.Emertimi, v.Nota
From Studenti S INNER JOIN Vlersimi V
    on s.Student_ID=v.Studenti
    INNER JOIN Lenda l
    on l.Kodi_L=v.Lenda
order by v.Nota desc
```

	Emri	Mbiemri	Emertimi	Nota
1	Lenonora	Peci	Sistemi i Bazave te te Dhenave	10
2	Liberta	Buja	Shkenca Kompjuterike 2 (POO)	10
3	Liberta	Buja	Rrjetat Kompjuterike	9
4	Edita	Syla	Arkitektura Kompjuterike dhe Organizimi	9
5	Visar	Dema	Sistemi i Bazave te te Dhenave	9
6	Kushtrim	Hoxha	Sistemi i Bazave te te Dhenave	8
7	Besiana	Shkembi	Shkenca Kompjuterike 2 (POO)	8
8	Edita	Syla	Rrjetat Kompjuterike	8
9	Liberta	Buja	Arkitektura Kompjuterike dhe Organizimi	7
10	Berat	Gashi	Rrjetat Kompjuterike	6

# LEFT OUTER JOIN

❑ SQL Server **LEFT OUTER JOIN** ose **LEFT JOIN** kthehen të gjitha rreshtat nga tabelat të shumta ku kusht i **JOIN** është plotësuar.

■ Sintaksa për **LEFT OUTER JOIN** në SQL Server (krijoj-SQL) është:

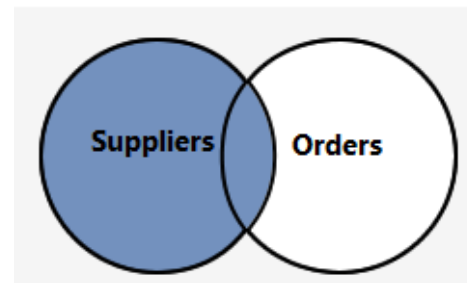
```
SELECT t1.kolona, t2.kolona
FROM tabela1 t1 LEFT JOIN
      tabela t2
ON t1.PRIMARYKEY = t2.FOREIGNKEY
```

S_ID	Supplier Name
1000	IBM
1010	Hewlett Packard
1020	Microsoft
1030	NVIDIA

O_ID	Suppliers_ID	Order Date
5000	1000	2003-05-12
5010	1010	2003-05-13
5030	1030	2003-05-14
5040	1000	2003-06-14

## LEFT OUTER JOIN

```
Select S.S_ID, S.[Supplier
Name], O.O_ID
From Suppliers S
LEFT OUTER Join Orders O
On S.S_ID =
O.Suppliers_ID;
```




Rezultati: **LEFT OUTER JOIN**

S_ID	Supplier Name	O_ID
1000	IBM	5000
1000	IBM	5040
1010	Hewlett Packard	5010
1020	Microsoft	NULL
1030	NVIDIA	5030

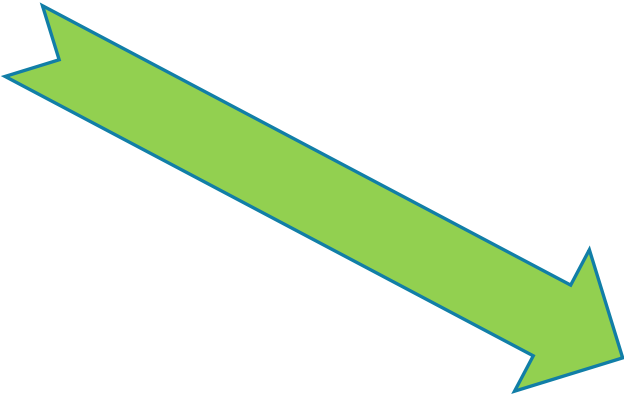
## shembull: **LEFT OUTER JOIN**

/\*Në raportin tuaj të shfaqeni të gjithë **studentet** që **nuk kanë kaluar ndonjë lëndë**.Në këtë raport të shfaqet emri, mbiemri, lënda dhe nota për të e verifikuar secilin student.\*/

```
Select s.Emri, s.Mbiemri , l.Emertimi, v.Nota  
From Studenti S LEFT JOIN Vlersimi V  
    on s.Student_ID=v.Studenti  
    LEFT JOIN Lenda l  
    on l.Kodi_L=v.Lenda  
where nota is null  
order by s.Emri asc
```



	Emri	Mbiemri	Emertimi	Nota
1	Liberta	Buja	Arkitektura Kompjuterike dhe Organizimi	7
2	Liberta	Buja	Shkenca Kompjuterike 2 (POO)	10
3	Liberta	Buja	Rrjetat Kompjuterike	9
4	Berat	Gashi	Rrjetat Kompjuterike	6
5	Edita	Syla	Rrjetat Kompjuterike	8
6	Edita	Syla	Arkitektura Kompjuterike dhe Organizimi	9
7	Visar	Dema	Sistemi i Bazave te te Dhenave	9
8	Kushtrim	Kollari	NULL	NULL
9	Antigona	Peci	NULL	NULL
10	Kushtrim	Hoxha	Sistemi i Bazave te te Dhenave	8
11	Lenonora	Peci	Sistemi i Bazave te te Dhenave	10
12	Besiana	Shkembi	Shkenca Kompjuterike 2 (POO)	8
13	Veton	Sopa	NULL	NULL
14	Alketa	Jakupi	NULL	NULL



	Emri	Mbiemri	Emertimi	Nota
1	Alketa	Jakupi	NULL	NULL
2	Antigona	Peci	NULL	NULL
3	Kushtrim	Kollari	NULL	NULL
4	Veton	Sopa	NULL	NULL

# RIGHT OUTER JOIN

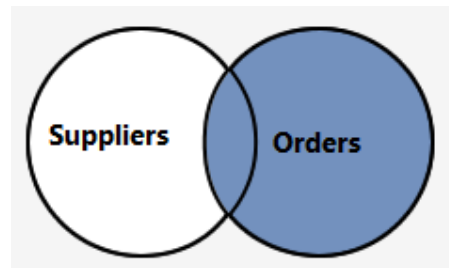
❑ SQL Server **RIGHT OUTER JOIN** ose **RIGHT JOIN** kthehen të gjitha rreshtat nga tabelat të shumta ku kusht i **JOIN** është plotësuar.

■ Sintaksa për **RIGHT OUTER JOIN** në SQL Server (krijoj-SQL) është:

```
SELECT t1.kolona, t2.kolona
FROM tabela1 t1 RIGHT JOIN
      tabela t2
ON t1.PRIMARYKEY = t2.FOREIGNKEY
```

## RIGHT OUTER JOIN

```
Select S.S_ID,
S.[Supplier Name],
O.O_ID
From Suppliers S
RIGHT OUTER Join
Orders O
On S.S_ID =
O.Suppliers_ID;
```



S_ID	Supplier Name
1000	IBM
1010	Hewlett Packard
1020	Microsoft
1030	NVIDIA

O_ID	Suppliers_ID	Order Date
5000	1000	2003-05-12
5010	1010	2003-05-13
5030	1030	2003-05-14
5040	1000	2003-06-14

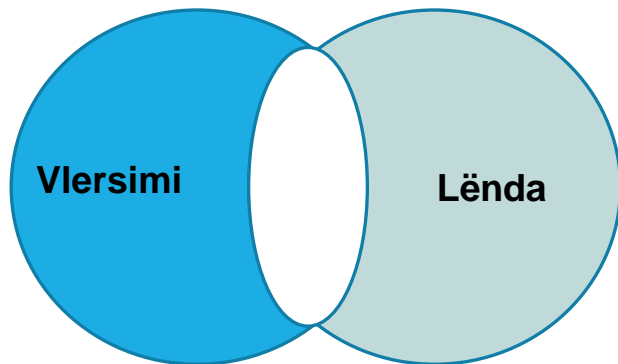
Rezultati: **RIGHT OUTER JOIN**

S_ID	Supplier Name	O_ID
1000	IBM	5000
1010	Hewlett Packard	5010
1030	NVIDIA	5030
1000	IBM	5040

## shembull: **RIGHT OUTER JOIN**

/\*Në **raportin** tuaj të shfaqeni Lëndët, ku nuk ka kaluar **ndonjë** student. Në këtë raport të shqet lënda, dhe ECTS për të e verifikuar secilin Lëndë.\*/\*

```
Select S.emri as 'Studenti', V.Lenda, l.Emertimi, l.ECTS
From Studenti S Left JOIN Vlersimi V
    on s.Student_ID=v.Studenti
    RIGHT JOIN Lenda l
    on l.Kodi_L=v.Lenda
where nota is null /*ose mundeni where v.Lenda is null*/
```



	Lenda	Emertimi	ECTS
1	CAO	Arkitektura Kompjuterike dhe Organizimi	5
2	CAO	Arkitektura Kompjuterike dhe Organizimi	5
3	CN	Rrjetat Kompjuterike	5
4	CN	Rrjetat Kompjuterike	5
5	CN	Rrjetat Kompjuterike	5
6	CS2	Shkenca Kompjuterike 2 (POO)	7
7	CS2	Shkenca Kompjuterike 2 (POO)	7
8	DBS	Sistemi i Bazave te te Dhenave	6
9	DBS	Sistemi i Bazave te te Dhenave	6
10	DBS	Sistemi i Bazave te te Dhenave	6
11	NULL	Alogoritme dhe Struktura e te Dhenave	4
12	NULL	Strukutura Diskrete 1	5

	Studenti	Lenda	Emertimi	ECTS
1	NULL	NULL	Alogoritme dhe Struktura e te Dhenave	4
2	NULL	NULL	Strukutura Diskrete 1	5

# FULL OUTER JOIN

❑ SQL Server **FULL OUTER JOIN** ose **FULL JOIN** kthehen të gjitha rreshtat nga tabelat të shumta ku kusht i **JOIN** është plotësuar.

■ Sintaksa për **FULL OUTER JOIN** në SQL Server (krijoj-SQL) është:

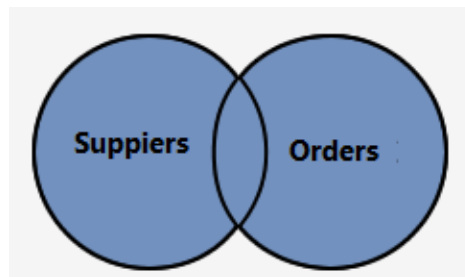
```
SELECT t1.kolona, t2.kolona
FROM tabela1 t1 FULL JOIN
      tabela t2
ON t1.PRIMARYKEY = t2.FOREIGNKEY
```

S_ID	Supplier Name
1000	IBM
1010	Hewlett Packard
1020	Microsoft
1030	NVIDIA

O_ID	Suppliers_ID	Order Date
5000	1000	2003-05-12
5010	1010	2003-05-13
5030	1030	2003-05-14
5040	1000	2003-06-14

## FULL OUTER JOIN

```
Select S.S_ID, S.[Supplier
Name], O.O_ID
From Suppliers S
FULL OUTER JOIN Orders O
On S.S_ID =
O.Suppliers_ID;
```



Rezultati: **FULL OUTER JOIN**

S_ID	Supplier Name	O_ID
1000	IBM	5000
1000	IBM	5040
1010	Hewlett Packard	5010
1020	Microsoft	NULL
1030	NVIDIA	5030

## shembull: **FULL OUTER JOIN**

*/\*Në raportin tuaj të **verifikon** cilit **student** nuk ka kaluar ndonjë **lëndë**, po ashtu në të njëjton raport të **shfaqeni** Lëndët në të cilat kalushmeria ka qenë zero. Në raport të shaqet *emrin, mbiemrin e studentit dhe lënda, dhe ECTS.*\*/*

```
Select s.Emri, s.Mbiemri, V.Lenda, l.Emertimi, l.ECTS
From Studenti S FULL JOIN Vlersimi V
    on s.Student_ID=v.Studenti
    FULL JOIN Lenda l
    on l.Kodi_L=v.Lenda
Where v.Lenda is null
```

	Emri	Mbiemri	Lenda	Emertimi	ECTS
1	Kushtrim	Hoxha	NULL	NULL	NULL
2	Agnes	Peci	NULL	NULL	NULL
3	Veton	Sopa	NULL	NULL	NULL
4	Alketa	Jakupi	NULL	NULL	NULL
5	NULL	NULL	NULL	Strukutura Diskrete 1	5
6	NULL	NULL	NULL	Alogoritme dhe Struktura e te Dhenave	4

	Emri	Mbiemri	Len...	Emertimi	ECTS
1	Liberta	Buja	CAO	Arkitektura Kompjuterike dhe Organizimi	5
2	Liberta	Buja	CS2	Shkenca Kompjuterike 2 (POO)	7
3	Liberta	Buja	CN	Rrjetat Kompjuterike	5
4	Berat	Gashi	CN	Rrjetat Kompjuterike	5
5	Edita	Syla	CN	Rrjetat Kompjuterike	5
6	Edita	Syla	CAO	Arkitektura Kompjuterike dhe Organizimi	5
7	Visar	Dema	DBS	Sistemi i Bazave te te Dhenave	6
8	Kushtrim	Kollari	NULL	NULL	NULL
9	Antigona	Peci	NULL	NULL	NULL
10	Kushtrim	Hoxha	DBS	Sistemi i Bazave te te Dhenave	6
11	Lenonora	Peci	DBS	Sistemi i Bazave te te Dhenave	6
12	Besiana	Shkambi	CS2	Shkenca Kompjuterike 2 (POO)	7
13	Veton	Sopa	NULL	NULL	NULL
14	Alketa	Jakupi	NULL	NULL	NULL
15	NULL	NULL	NULL	Strukutura Diskrete 1	5
16	NULL	NULL	NULL	Alogoritme dhe Struktura e te Dhenave	4



# SELF JOINS (relacione rekursive)

- ❑ Ne duam listën e punonjësve dhe emrin menaxherve të tyre. Menaxheri mund të jetë **çelësi i huaj** në tabelën e menaxherit, por duhet të jetë **çelësi i 'huaj'** në tabelën e punonjësve në këtë rast
  - SQL na lejon ta bëni këtë duke i dhënë **alias** :

```
SELECT Emp.Name, Manager.Name  
FROM Employees E, Employees Manager  
WHERE E.Mgr = Manager.EmpNbr
```

Employees (physical table)

EmpNbr	Name	Title	Mgr
105	Mary Smith	Analyst	104
109	Jill Jones	Sr Analyst	107
104	Sally Silver	Manager	111
107	Pat Brown	Manager	111
111	Eileen Howe	President	

Emp (alias/virtual)

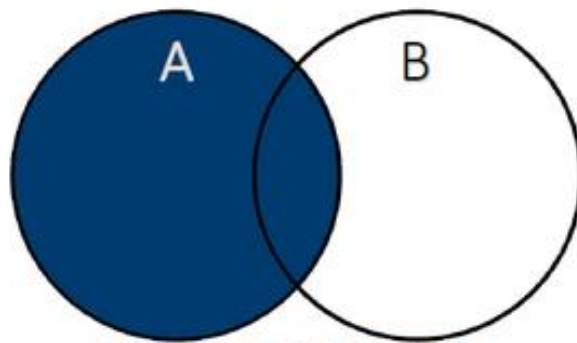
EmpNbr	Name	Title	Mgr
105	Mary Smith	Analyst	104
109	Jill Jones	Sr Analyst	107
104	Sally Silver	Manager	111
107	Pat Brown	Manager	111
111	Eileen Howe	President	

(Implicit syntax)

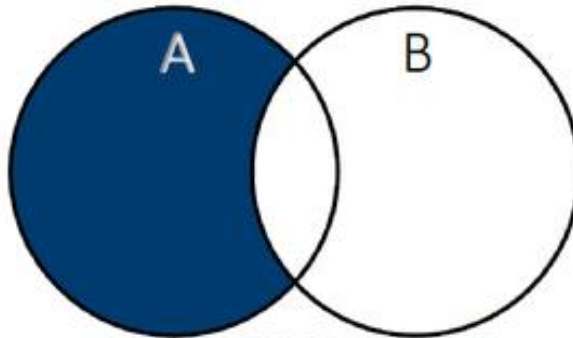
```
SELECT Emp.Name, Manager.Name  
FROM Employees AS Emp INNER JOIN Employees AS Manager  
ON Emp.Mgr = Manager.EmpNbr
```

Manager (alias/virtual)

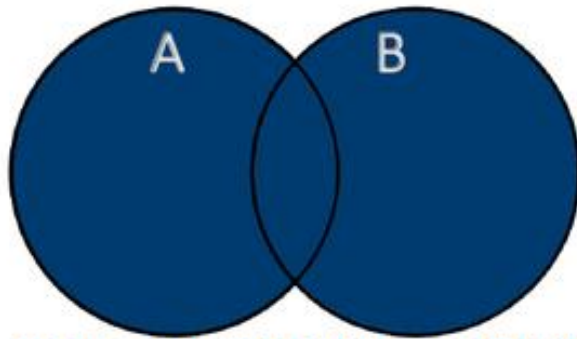
EmpNbr	Name	Title	Mgr
105	Mary Smith	Analyst	104
109	Jill Jones	Sr Analyst	107
104	Sally Silver	Manager	111
107	Pat Brown	Manager	111
111	Eileen Howe	President	



LEFT INCLUSIVE

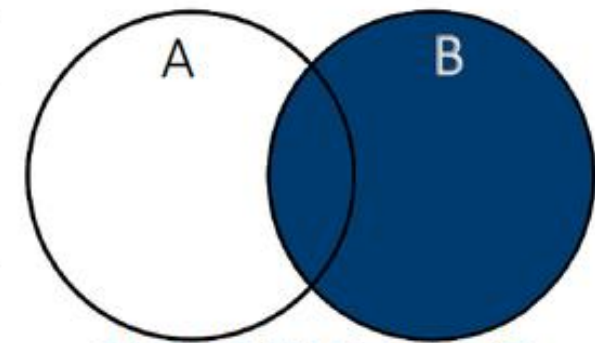


LEFT EXCLUSIVE

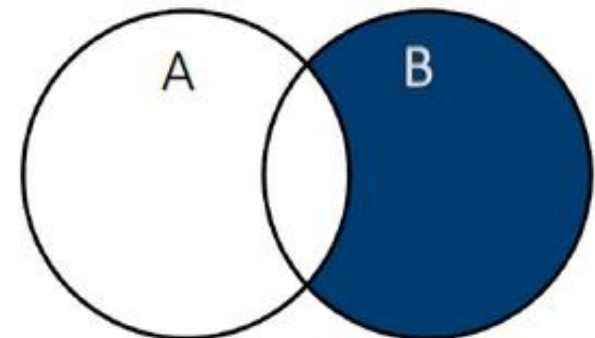


FULL OUTER INCLUSIVE

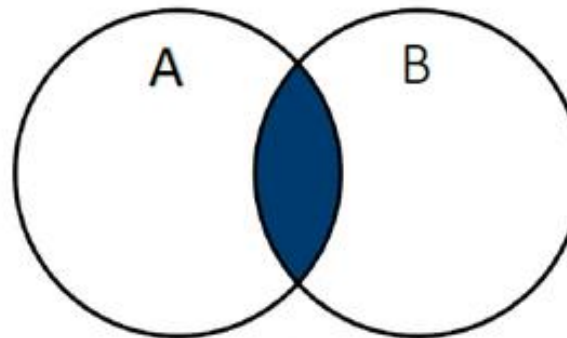
SQL JOINS	
<b>LEFT INCLUSIVE</b> SELECT [Select List] FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key	<b>RIGHT INCLUSIVE</b> SELECT [Select List] FROM TableA A RIGHT OUTER JOIN TableB B ON A.Key= B.Key
<b>LEFT EXCLUSIVE</b> SELECT [Select List] FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key WHERE B.Key IS NULL	<b>RIGHT EXCLUSIVE</b> SELECT [Select List] FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key WHERE A.Key IS NULL
<b>FULL OUTER INCLUSIVE</b> SELECT [Select List] FROM TableA A FULL OUTER JOIN TableB B ON A.Key = B.Key	<b>FULL OUTER EXCLUSIVE</b> SELECT [Select List] FROM TableA A FULL OUTER JOIN TableB B ON A.Key = B.Key WHERE A.Key IS NULL OR B.Key IS NULL
<b>INNER JOIN</b> SELECT [Select List] FROM TableA A INNER JOIN TableB B ON A.Key = B.Key	



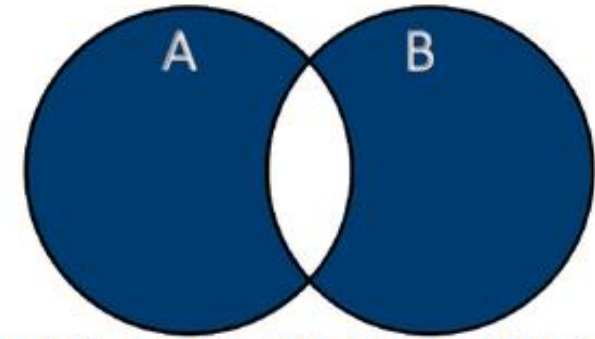
RIGHT INCLUSIVE



RIGHT EXCLUSIVE



INNER JOIN



FULL OUTER EXCLUSIVE

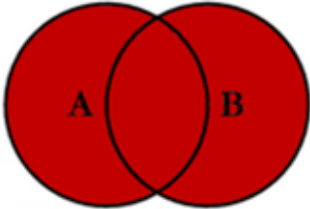
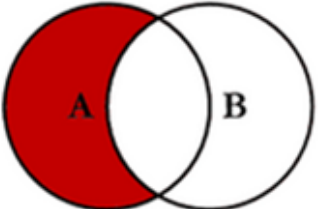
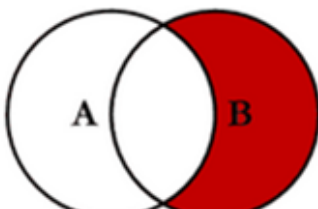
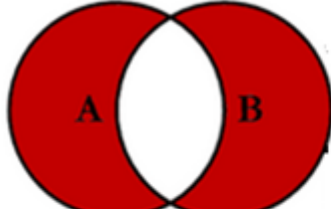
Table A = Employees ( EmployeeID, LastName, ... ) ^ Table B = Orders (OrderID, EmployeeID, ... )	
 <p>SELECT &lt;select_list&gt; FROM TableA A FULL OUTER JOIN TableB B ON A.Key = B.Key</p>	<p><b>All employees and all orders</b></p> <p>SELECT Employees.LastName, Orders.OrderID FROM Employees FULL JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID</p>
 <p>SELECT &lt;select_list&gt; FROM TableA A LEFT JOIN TableB B ON A.Key = B.Key WHERE B.Key IS NULL</p>	<p><b>Employees who have not registered any orders.</b></p> <p>SELECT Employees.LastName, Orders.OrderID FROM Employees LEFT JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID WHERE Orders.OrderID IS NULL</p>
 <p>SELECT &lt;select_list&gt; FROM TableA A RIGHT JOIN TableB B ON A.Key = B.Key WHERE A.Key IS NULL</p>	<p><b>Orders that were not registered by any employee.</b></p> <p>SELECT Employees.LastName, Orders.OrderID FROM Employees RIGHT JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID WHERE Employees.EmployeeID IS NULL</p>

Table A = Employees ( EmployeeID, LastName, ... ) ^ Table B = Orders (OrderID, EmployeeID, ... )	
 <p>SELECT &lt;select_list&gt; FROM TableA A FULL OUTER JOIN TableB B ON A.Key = B.Key WHERE A.Key IS NULL OR B.Key IS NULL</p>	<p><b>Employees who have not registered any orders and orders that have not been registered by any employee.</b></p> <p>SELECT Employees.LastName, Orders.OrderID FROM Employees FULL OUTER JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID WHERE (Employees.EmployeeID IS NULL) OR (Orders.OrderID IS NULL)</p>
<p><b>X</b></p>	<p><b>All possible combinations between employees and orders (Cartesian product)</b></p> <p>SELECT Employees.LastName, Orders.OrderID FROM Employees CROSS JOIN Orders</p>

[https://way2tutorial.com/sql/sql\\_join\\_types\\_visual\\_venn\\_diagram.php](https://way2tutorial.com/sql/sql_join_types_visual_venn_diagram.php)

<https://www.oracletutorial.com/oracle-basics/oracle-joins/>

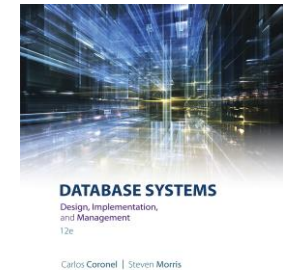
# Pyetje

---



# Referenca

**Kapitulli 7 & Kapitulli 8:** Database Systems: Design, Implementation, and Management, 12th Edition Carlos Coronel and Steven Morris.



**Kapitulli 3:** Database System Concepts, 7<sup>th</sup>ED – Abraham Silberschatz, Henry F. Korth, S. Sudarshan



<https://www.techonthenet.com/sql/update.php>  
<https://www.zentut.com/sql-tutorial/sql-update/>

# DDL: shembull 2

---

## Shembull: Baza e të Dhënave Relacionale

Kosumatori					
kosumatori_ID	emri	mbiemri	rruga	kodi_postar	qyteti
1	Rron	Berisha	27 Deshmoret 1912	10000	PR
2	Edona	Gashi	76 Hasan Mekuli	50000	GJK
7	Alton	Krasniqi	125 Skenderbeu	20000	PZ
26	Reneta	Musa	82 Bujaria	30000	PJ
8	Erlis	Sopa	67 Bajrush Doda	10000	PR
9	Antoneta	Gerxhaliu	10 Hasan Prishtina	10000	PR

Furnizuesi		
Furnizuesi_ID	emri_kompanis	qyteti
A50	comtrada	PR
X88	alltech	GJK
X90A	hitech	PJ

Produkti				
nr_serik	emri_brandit	tipi	çmimi	viti_prodhimit
G62-465DX	HP	laptop	630.59	2014
23-g040xt	HP	All-in One	890.99	2016
W88401231AX	apple	laptop	1360.89	2016
266-SKX-X1	Lenovo	laptop	1170.96	2017

Urdhresa					
urdhresa_ID	kosumatori_ID	nr_serik	data	sasia	statusi
100	1	G62-465DX	14.07.2017	1	dorëzuar
200	7	23-g040xt	19.10.2017	2	hapur
300	1	W88401231AX	20.10.2017	1	dorëzuar
400	9	266-SKX-X1	2.11.2017	3	hapur

skemat e relacioneve përkatëse janë si më poshtë:

Kosumatori (kosumatori\_ID, emir, mbiemri, rruga, kodi\_postar, qyteti)  
 Produkti (nr\_serik, emir\_brandit, tipi, çmimi, viti\_prodhimit)  
 Urdhresa (urdhresa\_ID, kosumatori\_ID(FK), nr\_serik(FK), data, sasia, statusi)  
 Furnizuesi (furnizues\_ID, emir\_kompanis, qyteti)



# Dëfinimet e tabelave

```
--Krijon Bazen e te Dhenave
create database L_DDL1718
create table Kosumatori
(
    kosumator_ID int Primary key,
    emri varchar(50) not null,
    mbiemri varchar(50),
    rruga varchar(100),
    kodi_postar varchar(10),
    qyteti char(5)
)
create table Produkti
(
    nr_serik varchar (20),
    emri_brandit varchar(50) not null,
    tipi varchar(20),
    çmimi decimal,
    viti_prodhimit date,
    primary key(nr_serik)
)
```

```
create table urdhresa
(
    urdhresa_ID int primary key,
    kosumatori int,
    produkti varchar(20),
    date_urdhreses date,
    sasia decimal(4, 2),
    statusi varchar(10),
    foreign key(kosumatori) references
        Kosumatori(kosumator_ID)
    ON Update cascade On Delete set null,
    foreign key(produkti) references
        Produkti(nr_serik)
    ON UPDATE CASCADE,
)
```